Erin Farley: All right, I think we'll go ahead and get started. Good afternoon, everyone. My name is Erin Farley. I am one of JRSA's research associates. For those of you who are less familiar with JRSA, it stands for Justice Research and Statistics Association. We are a national nonprofit organization dedicated to the use of research and analysis to inform criminal and juvenile justice decision making. We are comprised of a network of researchers and practitioners, which at the core includes directors and staff from state statistical analysis centers.

Erin Farley: It is my pleasure today to welcome you to a webinar on web scraping. This will be presented by Matthew Stubenberg. He graduated from the University of Maryland in 2013 and has created several web scrapers and the resulting data has been used to win class action lawsuits, find houses with higher water bills and expunge criminal records. He has won several awards, including the ABA's On the Rise top 40 young lawyers and the Maryland Bar Foundations Award for the Advancement of the Rights of the Disadvantaged. Matthew is currently the IT director and the staff attorney at the Maryland Volunteer Lawyer Service and continues to strive to merge technology and the law for the benefit of low income individuals. Welcome, Matthew.

Erin Farley: Before we go any further, I do want to thank our partners at the Bureau of Justice Statistics for helping to make this webinar possible. I would also like to cover a few logistical items. Okay, we will be recording today's session for future playback. The link to the recording will be posted on JRSA's website and emailed to attendees and those on the waiting list. Today's webinar is being audio cast via both the speakers on your computer and teleconference.

Erin Farley: If you have speakers on your computer or headphones, and you are not the presenter, we recommend listening to the webinar using your computer speakers or headphones. To access the audio conference, select audio from the top menu bar and then select Audio Conference. Once the audio conference window appears, you can view the teleconference, call in information or join the audio conference via your computer.

Erin Farley: Okay, if you have any questions and you want to communicate with JRSA staff, we encourage you to submit them using the chat feature on the right side of your screen. Please select, if you select [inaudible 00:02:55] and you can see, you can go all the way down to All Participants, that way we can see, everyone can see all the questions.

Erin Farley: This session is scheduled for about an hour. If you do have any technical difficulties or get disconnected, just try and reconnect the way you originally have. If you still find that you are having problems, you can reach out to our tech guru, which is Jason Trask and he is available at Jtrask@jrsa.org.

Erin Farley: In the last five minutes of the webinar, we will ask you to complete a short survey and this information helps us to plan and improve future webinars and to meet our reporting requirements.
Erin Farley: One last thing, we do have and are aware that some people use one computer but have multiple viewers, so we're trying to track the number of attendees. If you have a group of people that are watching this webinar, if you don't mind, if you could go to the chat window, type the name of the person registered and the total number of additional people in the room, that would really help us keep track of who's attending our webinars. We would greatly appreciate it.

Erin Farley: Again, if you have any questions as we're going through, just put them into the chat making sure to select all participants and as Matthew's sort of going through his presentation I will chime in here and there to ask him the question. I think, with that, I will pass it over to Matthew and we can get started. Welcome.

Matthew Stubenberg: All right, thank you. All right, can you see the screen okay?

Erin Farley: Yes I can.

Matthew Stubenberg: Perfect, perfect. All right, so thank you so much for having me. Again, my name is Matthew Stubenberg. I'm the IT director of a group called Maryland Volunteer Lawyer Service here in Maryland. I'm also a staff attorney. I always like to speed through these biographies, but just briefly, I graduate with a BA in Political Science from the University of Maryland Baltimore County in 2009 and with a law degree from the University of Maryland School of Law in 2013.

Matthew Stubenberg: The reason I bring this up is I'm primarily a self taught coder with a focus in kind of back end web development web scraping, so if any of this interests you, I wouldn't give up hope just because you didn't get a degree in computer, right? My degree was in political science. Anybody can learn this stuff. You don't have to be a savant from age 5.

Matthew Stubenberg: I'm going to be talking today about web scrapers, which is a really great way to gather data that exists out there but isn't in a form that you can actively analyze it or go through it. I have a couple of examples, but before I do I'm going to jump into exactly what a web scraper is.

Matthew Stubenberg: In order to go over what a web scraper is, I first have to kind of cover how using a website normally works. When you normally use a web page, you go to something like Google.com. Well you first open a browser and then you go to Google.com and let's say you're trying to find some great webinars. What you do is you type great webinars into the search box and click search.

Matthew Stubenberg: What your browser does is it will take that information, the fact that you typed great webinars into the text box, the fact that you clicked the Google search button and any other boxes or options you may have checked, your browser then will condense all of that into some computer code that it will then send to the server that's hosting the website.
Matthew Stubenberg: What the server does with that, is the server will say okay, it looks like this person is searching great webinars. They clicked the Google search box, not the I'm feeling lucky box, the settings are set for this. Google has kind of their magic software, they go out and they find the websites you're looking for. Ultimately what the server does is it then returns a webpage to be viewed on your browser and then that's how you get the pretty website with all of the great webinar links out there.

Matthew Stubenberg: That server gives the data back in a form called HTML. An HTML is a computer language that is what websites are built out of. HTML is designed to be displayed in a browser. It contains, you can kind of see a small snippet here in the bottom left hand corner what some HTML looks like. A browser reads that to figure out where the link should go, what color the text should be, things like that.

Matthew Stubenberg: That's how using the internet works normally on a web browser. How does building a web scraper differ from that? Well, the primary thing is that it cuts out the human interaction piece, right? You're not actually typing in great webinars or clicking any buttons. In fact, it removes the entire browser completely.

Matthew Stubenberg: The web scraper comes into this whole process at the point after where the web browser would collect all that data, put it into a kind of some packets designed to go to the server. What the web scraper essentially does is mimic the information that is sent from the web browser to the server.

Matthew Stubenberg: The server doesn't know whether it's a web browser that's sending it information, a web scraper or anything like that. When the web server gets the same, gets information, it processes it just like anything else. It'll get the exact same information sent to it from the web browser from the web scraper instead. The server will, let's say it's the Google server, it will do the same kind of analysis, it'll find the links to the websites that you want to visit and will return all of those results in the HTML format that it would normally, thinking that it's going to be displayed into a web browser.

Matthew Stubenberg: What a web scraper then does with that HTML, instead of displaying it, since there is no browser for it to be displayed in, it actually analyzes and scans through that HTML looking for specific things. The web scraper is usually designed to look for something specific. I have some examples of how to do that. It will scan through the HTML looking for whatever piece of information it wants and then save that information into a database.

Matthew Stubenberg: There's some really useful applications for this type of web scraper here. There's a number of really popular web scrapers out there that might kind of make a little bit more sense out of this.
Matthew Stubenberg: The biggest and most popular web scraper by far is the Google web scraper, right? When you go to Google and you type in baseball, how does google actually know all the websites that contain the words baseball or that are about baseball? Well there's a number of different ways, but the primary way is that Google has web scrapers that scour the entire internet. Let's say you start a webpage about baseball, Google will eventually find your webpage with their web scraper. It will scan all the contents of your website and it'll say hey the word baseball here is listed 45 times and there seems to be a lot of other indicators that this is a website about baseball. Let's save that in the database.

Matthew Stubenberg: Then when you search the word baseball, it goes to the Google database and says hey we have these ten websites that all mention baseball. Here are the websites they probably want to go to. That's one of the most popular and kind of biggest uses of web scrapers, at least in a commercial sense when it comes to Google.

Matthew Stubenberg: Then if you've seen the sites like Trivago or Priceline or Kayak where they will compare a lot of different options all at once. You can find the cheapest hotel, right? Well the way these sites do that is you go to their website, you say hey I'm trying to find a hotel between March 1st and March 3rd in Miami. What's the cheapest hotel rate? What these websites will do is using a web scraper, it'll take the data that you gave the web scraper and go to all of the hotels that it has listed in Miami and using that same time frame, March 1st to March 3rd, see what the prices are that come back. It then goes through it all and says here are the five cheapest options.

Matthew Stubenberg: Finally, just as another kind of example of a commercial web scraper is Venmo, which is a popular money transfer app on a lot of phones. What you do is you connect Venmo to your bank account. Interestingly a lot of banks don't have nice API's to connect everything. What you need in order for Venmo the app to be able to pull money out of your account or put money into it, is they actually need your credentials which is why you give them your user name and password. Using a web scraper, they then go to Capital One or Wachovia or something and almost pretending to be you, log into your account using a web scraper and then they have pre-programmed instructions on how to pull out money, how to send money and the like. There's a lot of different programs people use on a daily basis that all rely very heavily on web scrapers.

Matthew Stubenberg: That kind of gives some background information as to how they've been used in the past. This is for, kind of research and statistic purposes, so I have two big examples that we accomplished here at [inaudible 00:12:57] that analyzed some public data using web scrapers and some statistics that didn't exist before.

Matthew Stubenberg: The first one I want to go over here is some water bill data that we were able to collect. Usually when I tell this people say why is collecting water bill data important at all? Well, in Baltimore City water bills over $750 will result in a tax sale of the house in order to pay for that water bill. If you're an elderly person
and let's say you own your house outright, and let's say unfortunately your spouse recently passed away and they were the ones who took care of paying the water bill. You lose track of it, the water bill starts to add up and maybe you're on social security and so it becomes very difficult to pay back a bill of $500, or $600, or $700, or $800.

Matthew Stubenberg: Then what that puts you at risk at is losing the house that you have already paid off just to pay off an $800 water bill. This is something that typically plagues a lot of people in Baltimore City that we wanted to try to prevent.

Matthew Stubenberg: Baltimore City has all of the water bill information public. How they make it public is they don't have it in a downloadable database. You can't just download an Excel file of everybody's water bill, but what they do have is they have a website where you can type in an address and you can get the current water bill. Here we have, this is the address I lived when I was in law school. If we type it in here, this is the information of the web page that is returned.

Matthew Stubenberg: You can see one of the pieces of information here that's displayed is the current water bill balance. It looks like my old landlord owes $360 on his water bill. The question we were trying to answer is because this data is publicly available, is there a way to figure out how many houses are at risk of being sold at tax sale to pay a water bill? On top of that, how many people are close to that line? How many people would be affected if legislation changed to move that number up or down some? This was data that didn't exist anywhere before.

Matthew Stubenberg: What we needed to do in order to get all this data is to use a web scraper. The first thing we needed to do is if you noticed on the website you could search by address number. We had to find a list of every address in Baltimore City in order to feed it into this web scraper we were going to create.

Matthew Stubenberg: Luckily and conveniently for us, Baltimore City did publish a giant Excel sheet of all of the property taxes in Baltimore City, which conveniently contained the addresses for every property in Baltimore City. Here's a kind of a screenshot of that. It contained 238,000 addresses. This made life kind of significantly easier for us.

Matthew Stubenberg: You can see there's a whole bunch of addresses here. It goes on for 238,000 rows. Now we had that piece of the puzzle solved. We had a list of all the addresses. The next piece was to actually build the web scraper.

Matthew Stubenberg: The steps that go into building a web scraper aren't too difficult. What you have to do is you have to see what the information looks like as it goes from the browser, when you do a normal search of a house to the server that hosts that website. In this case the water bill website. That's exactly what I did, except I used the Chrome browser and Chrome, most browsers have some kind of tool and Chrome is no different, has a tool that you can use to monitor what kind of traffic is going between the browser and the server.
Matthew Stubenberg: I activated that tool. It comes pre-installed on any browser, so anybody can do that. I did a normal search. I typed in that same address that I already knew and saw what kind of traffic was sent from the browser to the server. You can kind of see pieces of it right here. It was more than one page, unfortunately.

Matthew Stubenberg: All of this information is all packed into kind of one condensed computer code and is then sent over to the server. That's what the server uses to determine what address I'm trying to look up. By going through all the information that is sent to the server, there was one particular variable name here that had the address that I typed. You can see it here at the bottom of your screen. It's this very long CT$100 sign and it goes on and on and on and on and on. You can see the text here is 834 Holland Street, which was the address that I typed in.

Matthew Stubenberg: Using that variable, I could now build a program that simply added whatever I wanted to that variable. Instead of 834, we could replace it with 123 Main Street or whatever address we deemed fit.

Matthew Stubenberg: On the right hand side is some of the code that I used to actually build the web scraper. You'll notice it's actually not that much. These web scrapers, they take a little bit of time but they're not complicated where you need teams and teams of people to build them. If you are curious, all this code is available online so anybody can go online and see how to scrape the Baltimore City water bill website.

Matthew Stubenberg: What this code does is it mimics just enough of the information that comes from the web browser to make it look like it's a request from a web browser. You don't actually have to copy everything, you just need to copy enough because the server doesn't necessarily look at everything.

Matthew Stubenberg: With some trial and error I figured out what the server was looking for and that's the stuff that I duplicated. A lot of this is just kind of information that was sent that is kind of not really relevant, and then if you kind of squint your eyes since it's kind of small, you can see that this variable right here is the same variable that contained the address. We're able to then put in any address that we want and send that information off to the server.

Matthew Stubenberg: The server reads it as if it's any browser looking up any kind of address and then we'll send back the HTML for our program to display. Now that's kind of where the next step comes in since we're not actually using a browser, right? There's no HTML. There's no place for the HTML to be displayed, so what we have to do is we have to take a look at the HTML and figure out where this current balance is, where it is in the HTML.

Matthew Stubenberg: The HTML contains everything that you can see on your browser. We look through the HTML and low and behold, there was this one tag right there, it's a span tag that contained the value $360.32. This particular project was made very convenient for us because it had an ID in the span tag, with this really long
kind of thing and it ended with current balance, so we knew we were on the right track.

Matthew Stubenberg: What an ID tag lets you do is find that tag very quickly and easily using a computer program. It's not always as easy as that. We were able to use this code right here, which is what we built to basically, as the HTML comes back from the web scraper, look through all that HTML and find the one span tag in the HTML that has this ID. They can get the value of whatever's in between the two span tags and that will be whatever the current balance is.

Matthew Stubenberg: Just to get the current balance wasn't actually all that difficult to do. At that point we had the web scraper that would take an address, go to the water bill website, get the HTML, look through the HTML or parts of the HTML is the proper terminology and find the current balance. Then all we needed to do in order to pull out all of the water bill information is just put it into a giant loop.

Matthew Stubenberg: We had this giant Excel sheet. We built another small program that basically just passed one address at a time to our web scraper. The web scraper would take the address, add it to that information that gets sent to the server, retrieve the HTML that the server sends back, look through it, get the current balance and then the final piece was just to save that number and that address to a database so we could look it up later.

Matthew Stubenberg: We ran this 238,000 times, one for each address. It took probably the better part of a day for the program to complete all of these cycles, but at the end of it, we had something really interesting. We had a snapshot of what all of the water bills in Baltimore City looked like and where they were.

Matthew Stubenberg: This was data that didn't exist anywhere but was technically public. It kind of, now that we had this data, what exactly do we do with it, right? We were able to figure out a couple of things. First we found 13,000 addresses with a water bill over $750. These are all the houses that are at risk of tax sale, that will actually go to tax sale next year. We could then potentially reach out to them and say look, at some point this is going to come to bite you, let's take care of it now before the tax sale process gets started.

Matthew Stubenberg: We were also able to geocode all the addresses and then put all those addresses on a map so we created a nice heat map where you can see where the biggest problem areas lie. Then because we geocoded it we were actually able to, down to the house level, figure out where all of these houses with really large water bills that were going to have their house sold at tax sale existed.

Matthew Stubenberg: If you look at this map in the bottom right, we could have actually driven up to this neighborhood and gone to these four or five houses that had the red dots, which indicate a water bill over $750, knocked on the door and say hey, we're from MBLS and don't ask how we got your water bill, but we did and we're here
to offer some assistance. We can make a payment plans, we can challenge the water bill, there's all kinds of things.

Matthew Stubenberg: We've also been able to send letters to houses in poor zip codes that have water bills that are over a certain amount saying hey you're at risk here, MBLS can help, here's how we can help. Just as another kind of, I gave the example of elderly people sometimes getting into trouble with water bills, but another example is if you're a renter and your landlord isn't paying your water bill, that house is still sold at tax sale. Then, with some special circumstances, otherwise, if that house is sold at tax sale, you then get kicked out of that house, so it can still be a problem for you even if you're not the homeowner.

Matthew Stubenberg: You might think your landlord is paying those water bills, when in fact, they're not. This can be something that can really detriment a lot of people very quickly.

Matthew Stubenberg: All right, so that was the first kind of web scraper example that I went over. I just want to stop to see if there's any questions here in the chat room, or if there's anything about the water bill stuff before I go into the next piece here.

Matthew Stubenberg: Looks all quiet, alright.

Erin Farley: Yup, I don't see any questions at this time.

Matthew Stubenberg: All right. The next big, I see a link here. The question is can a web scraper get past CAPTCHA'S and the answer is sometimes. A CAPTCHA and I talk about this a little bit later, so I'm not going to go too much onto it, while there are sometimes ways to get around CAPTCHA'S, usually you probably shouldn't. CAPTCHA’S are a pretty good way of telling you that the website doesn't want to be scraped, and so, you know, you take some risks when you purposely try to go around their protective measures, but I talk about that in a little bit.

Matthew Stubenberg: The second big example we have here for web scrapers is with court data. In Maryland, we have a public website that the government runs here where anybody can look up anybody's criminal or civil cases. It contains a vast wealth of information, million of cases going back to about 1985, civil and criminal and traffic, all kinds of information here. It's a real wealth, but there's no way to download the information. All you can do is search by the parameters they give you. Here are the parameters, if you can kind of see there.

Matthew Stubenberg: You can search by someone's name and you can search by case number, but if you wanted to find all of the people charged with marijuana possession who were convicted, who lived in this zip code, this website doesn't have the parameters for that, so you can't do it. What we decided to do was build a program that scraped all of the data out from case search and then we would be able to search by whatever parameters we wanted to.
Matthew Stubenberg: You'll notice that one of the parameters was to search by case number. If you type in a case number it will load up the information for that particular case and it'll take you to that website.

Matthew Stubenberg: For instance, we type in a case number here, this is a random John Smith that I found and this is their criminal case. As you can see, it has a wealth of information for addresses, to tracking numbers, what they were charged with, what they received. This is how I approached the project is that we were going to build a web scraper that scraped this criminal case by case number.

Matthew Stubenberg: We built yet another web scraper. We followed a very similar method where we had a case number that we already knew, we activated this tool on Chrome in order to monitor the traffic that gets sent between the browser and the server and we decided to mimic it.

Matthew Stubenberg: Here's kind of a screenshot of the information as it gets sent from the browser to the server and you'll notice the important information is right here. This is where the case ID is the case number. We were then able to build another program where it mimicked the data and it left the case ID as a variable that we could swap in with a new case number.

Matthew Stubenberg: Once we got that, we got the HTML back for that particular case and then we had to build another little kind of piece of code here to go through all of the HTML and find all of the charge information, what they were charged with, what their name is, what their birth date is and kind of parse it out from the HTML so we can save it into a database.

Matthew Stubenberg: This piece was significantly harder than on the water bill website. On the water bill website we had a nice, easy ID tag that said this is where the current balance is. It only took one line of code to find that value. It was not so easy with the case search information. Here's kind of just a little screenshot of it, but you can see there's no ID tag. It just has a class tag, and classes are used to display the data in different ways. Either you can have a class that displays the text as red or bigger or smaller or things like that, but ultimately that didn’t help us too much.

Matthew Stubenberg: We had to build a very complex program here that was able to parse through all of the HTML and realize that this is the, what they were charged with, this is what they received, this is how many years they got in prison. We were able to do it. Then we built just another piece to save it to a database which is fairly straightforward.

Matthew Stubenberg: How did this loop work? Last time, we had a list of every address in Baltimore City that we could easily loop through. We didn't have a list of every case number because that didn’t exist. In fact, we tried to go that route, I did a FOIA request to the judiciary to try to get a list of every case number, but ultimately they deemed that project to be too time consuming and the cost would be too high, so they denied that request, so we had to find another workaround.
Matthew Stubenberg: What we did is we started with the earliest case number we could find. We went on case search manually and just typed in some information to try to find some cases from 1985. We found a case number from 1985 and that became our starting case number.

Matthew Stubenberg: We primed the web scraper that we'd just built with that first case number. The web scraper took that case number, went to the web site, pulled the HTML for that particular case back, parsed through it all to get the description and the verdicts for all the charges and the defendant information and saved it all into a database. Then, once it was finished saving into a database, we simply added 1 to the end of the case number and then gave that next case number back to the scraper.

Matthew Stubenberg: In this way we were able to keep going forward just adding 1 to the end of it until we ran out of case numbers, until we hit so many kind of, we programmed the web scraper to be able to identify if the case was not findable. If we searched a case number and nothing came up, then we knew that something was off. When we hit a certain number of those, we realized that we had reached the end of, we had reached the current day, so there was no more case numbers.

Matthew Stubenberg: That worked fairly well. We started from 1985 and we went all the way up to 2017 and we caught millions and millions and millions of cases. If we take a look at the total numbers here, we captured five million criminal cases, two and a half million civil cases. For civil we only went back to about 2010, over three million distinct people throughout all the criminal cases here and it wasn't just defendants. We had information now on three million people, including defendants, attorneys, witnesses, police officers, anybody that might be related to a case, we now had a lot of data on.

Erin Farley: Matthew?

Matthew Stubenberg: Yeah go ahead.

Erin Farley: Quick question, so I see in the case numbers that there's the B, and so I was wondering in some case numbers that letter will vary? You could have two letters depending on what sort of database or court database you're searching, did you run into that problem at all and how did you resolve that?

Matthew Stubenberg: So you're correct. I simplified a little bit of the case number incrementing that we did. It took a lot of staring at lots of case numbers to figure out exactly how they're all incremented. By adding one, I kind of simplified it. The B in the case number that you can see right here stands for Baltimore. Each county has it's own letter, in order to scrape every county, we actually had to, we scraped all of one letter, then we went back and scraped all of the next and all of the next until we captured all 24 counties here in Maryland.
Erin Farley: Okay.

Matthew Stubenberg: Then you're right, while this system worked fairly well for district court, even though there were a couple of pockets where, as courts transition from one thing to another and for some reason it would just skip 1,000 cases or something like that, so it did take a little bit of trial and error, but where we really ran into trouble was some of the circuit courts in Maryland. We captured their data as well, but circuit courts, historically in Maryland have a lot of autonomy and so they each had different case numbering systems.

Matthew Stubenberg: We had to build a different incrementer for, one for Baltimore City Circuit Court, one for Montgomery County Circuit Court, one for Prince George's County Circuit Court and that added an extra kind of wrench into our whole system.

Erin Farley: Right.

Matthew Stubenberg: It was something that was surmountable.

Erin Farley: I mean, that's a great example because, yeah, it seems like in some places you have to do, just overall you have to do sort of an initial assessment of how complex the numbering system or the lettering system is and then it could be as easy as adding one, or possibly multiple letters over different years or different jurisdictions and things like that, okay.

Matthew Stubenberg: Right and you will be surprised that once you start kind of diving into how a case number is incremented, just how few people actually know anything about it. We called a lot of courthouses to try to talk to a lot of clerks.

Erin Farley: Yeah.

Matthew Stubenberg: They all looked at me and said we have no idea, the system just generates the next case number.

Erin Farley: Wow.

Matthew Stubenberg: You finally have to find that one person in a back room somewhere who built it 20 years ago to try to help you explain it.

Erin Farley: That sounds about right. That sounds about right. All right, thank you.

Matthew Stubenberg: Yeah, yeah. Good question. Now that we had all of this data that we captured on millions of cases and we made sure to capture every single piece of information that was on Case Search, which is the name of the website here, so everything you could see was captured and stored in it's own field so we could search by it.
Matthew Stubenberg: You'll noticed that the race of the defendant, the zip code, what they were charged with and what they received are all part of what's on Case Search. With these four factors, we were able to actually create a little kind of interactive website where somebody could type in a particular crime in a particular zip code, pick a certain year and see all of the race breakdown for everyone who was convicted of a charge in that particular area.

Matthew Stubenberg: Of course, the statistics always kind of skew racially, and it's no different in Baltimore, and that was eventually [inaudible 00:36:28], our data wasn't actually used for it because we did it after the fact, but other data was used to show that, you know, despite African Americans probably making up a certain percentage in Baltimore City, they overwhelmingly received a majority of the marijuana citations or charges.

Matthew Stubenberg: The police didn't have a good explanation for that and eventually it led to the decriminalization of marijuana here in Maryland. This data can have really impactful results. This tool kind of helps dive into it a little bit further for historical purposes to figure out exactly, how many zip codes, what was the race breakdown of a particular charge? This is just what we chose. You could use this data to do almost anything, even height and weight are on here. You could see how tall is the average marijuana user? You can really get some really interesting data out there.

Matthew Stubenberg: We also captured all the civil cases and we built this interactive tool that lets you look up the number, or a particular plaintiff and see how many cases they had in a particular year and what the total judgments were for that particular year.

Matthew Stubenberg: In Maryland here we have a group called Big Boys Bail Bonds that does, that sues a ton of people. In 2013, we were able to say okay they were involved in 1700 cases in just that one year and received judgments of over 11 million dollars. This was information that is not lookupable, if that's a word, anywhere else. You couldn't do a Freedom of Information request for it, you couldn't ask the courts for it. We're not even sure if Big Boys Bail Bonds might have this information.

Matthew Stubenberg: It put a lot of data kind of at the fingertips that didn't exist before.

Matthew Stubenberg: We were able to take this and actually go a little bit further and add some value to the data that we collected. I'm an attorney and I do a lot of criminal record expungement. Expungement is the removal of a criminal case from someone's criminal record. In Maryland, conveniently, all of the factors you need in order to determine whether a case is eligible for expungement or not is on Case Search., is on that case when you load up your case number, you can determine whether that case is eligible for expungement or not.
Matthew Stubenberg: I created a program that used kind of branch logic and different kind of algorithms to determine, based on the expungement law here in Maryland, to determine whether a particular case is eligible for expungement. We ran that, once we developed that algorithm that just took a few inputs, right? It takes what you were charged with, what you received, when you received it, things like that. We ran that algorithm on every case that we had just downloaded. What were we able to do with that?

Matthew Stubenberg: We suddenly had a list of every criminal case in Maryland that was eligible for expungement, which never existed before. We geocoded a bunch of the addresses and we were able to create a map here in Baltimore City of every defendant's house with an eligible case for expungement.

Matthew Stubenberg: This is kind of a cross section of Baltimore City and each one of those pins is a person with an eligible case for expungement. We were able to use this map to better target resources. We sent out mailers to people. When we hold clinics, we used this map to figure out what the impact will be or how to predict how many people will need that service in a particular neighborhood and then we can adjust to how many attorneys we need to send in order to help with the load of people.

Matthew Stubenberg: This information can also be used from a policy standpoint to advocate for better expungement rules, for [inaudible 00:40:27]stage legislators who don't know how big the problem is in their district can now go in and say hey, wow this, there's a lot of people in my district who would benefit from expungement.

Matthew Stubenberg: Then we can also actually look up what the effect of certain laws would be. Because there's no state sponsored database of expungeable cases, nobody has any idea of if we added, let's say the charge of theft to the expungement list, how many people would that help? How many cases would that make eligible? How many extra staff members at the court would we need to hire to deal with the uptick in people?

Matthew Stubenberg: It provides a lot of really powerful analytics and statistics that didn't exist before.

Erin Farley: Matthew, I think we had a question. This might weave into a little bit later, but how do you get to the web scraper in Google was a question?

Matthew Stubenberg: How do you get to the web scraper in Google?

Erin Farley: I wonder if that's-

Speaker 3: You were saying that Google has a setting that you could turn on and look and see how it translated the code?
Matthew Stubenberg: Oh I see, yeah, yeah. If you're using Chrome, I can show you here on a, has my browser popped up?

Speaker 3: Yep.

Erin Farley: Yes.

Matthew Stubenberg: Okay, so this is just a regular Google page. For here, if you go up, it's in this, or the easy way to do it is if you right click anywhere and you click Inspect or Control Shift I, it will bring up this whole kind of tool right here. It starts out in the Elements tab where you can look through and find out all of the different elements in the web page you're looking at and this is all the HTML. Then if you click something called Network, this will monitor all the network that gets sent from your Chrome Browser to whatever server you're interacting with.

Matthew Stubenberg: If we then, you know, great webinars, all of this data comes up and if, just for simplicity purposes, this first one is usually what you're looking for because that holds the bulk of the data. When I ran that search, this is what was sent to Google, so you can see it contains, here's some cookie information because Google tracks everything, right? Here's my search phrase, all of the information I used. What you can do is you can try to mimic all of this information into a program and, except just replace what some of the parameters are in order to find out your own kind of searches.

Matthew Stubenberg: This is very simplistic. Websites work in a number of different ways, not everything is easily scrap-able. There's a lot of JavaScript that can asynchronously pull data from different sources. It can complicate the whole process. Some sites are more suited for scraping than others. Hopefully, did that sound like an answer?

Erin Farley: Yeah.

Speaker 3: Yes.

Erin Farley: Thank you.

Matthew Stubenberg: All of the browsers have one. Firefox has one, I'm sure Internet Explorer has one. Okay, so the two examples I've used so far are about multi web page lookup tools or kind of using it in bulk to pull out an entire data set, but much like the kind of Trivago or the Venmo app, single based web scrapers, web scrapers that just are meant to look for one thing and return it back and then just wait until it's called again are also really powerful so I want to just go over two of the projects that we did with that just to show you the other side of web scrapers when it comes to nonprofits.

Matthew Stubenberg: The first one is a website I created called MDExpangement.com. What this is, is it's kind of a, it combines all of the different aspects that we talked about before
with the Case Search lookup. Because you can look up, we used the web scraper we developed to look up a case on Case Search, added that case number to pull back the HTML, put all of that HTML into our algorithm. Then what we did is we created a website where you could actually type in your case number.

Matthew Stubenberg: You’re John Smith, you just went to a court hearing and your case was thrown out or somehow ended and you want to see if it’s eligible for expungement. What you can do is you can now to MDExpugnement.com, type in your case number and on the fly, the web scraper will take that case number, get the most recent data for it and pass it through the algorithm to determine if it’s likely eligible for expungement. This all takes a fraction of a second.

Matthew Stubenberg: Then, what the website does is down here if it’s likely eligible, it will tell you likely eligible for expungement. We actually then took it a step further, which isn't [inaudible 00:45:40] but we populated all the forms you would need in order to file for expungement. In about five seconds, knowing nothing more than your case number, you can actually have the forms you need to file with the court in order to get your criminal case expunged.

Matthew Stubenberg: It's been very helpful and so far we've expunged, with the website, over 34,000 cases in the last two years.

Erin Farley: Matthew, real quick, while you’re switching sides, there was a question, just sort of a general language question. What language browser did you use? I’m partial to Beautiful Soup in Python 3 and Selenium Phantom JS, I think that's what that says, but curious to know what you like.

Matthew Stubenberg: Yeah, yeah, so that’s a great tool. It's an advanced question, there. I am like you, I'm partial to Python. I’ve built these web scrapers in almost everything. A lot of it is PHP, which is, in my mind, a really great language for web scraping if you don't plan to do a lot of them. If you plan to run hundreds of thousands of them, you need to run a programming language that can do threading very well and something like Python is then better.

Matthew Stubenberg: I've built them in Python and Java and PHP. I have used Phantom JS as well. Phantom JS is, in my mind, a little bit more complicated to use and it can add on some significant time. If you're running a million cases, that extra second that it takes for each web scraper to pull can add on an extra month of processing power. Just for everyone on the audience, there’s two kind of main pieces there. One is making a request kind of in a one fell swoop kind of way, which is usually what Python or PHP will do if you use a [inaudible 00:47:45] or something.

Matthew Stubenberg: What Phantom JS does, it actually more mimics how a human would interact with a website. For certain websites, not all of the information will be displayed at once. Some of the information will populate, especially if it's a lot of information, what the website will do is it will initially load the frame to hold all
of the information and then bring the information to different pieces of that web page as it all loads, that way the person isn't waiting to see the website until everything's loaded. It's able to display it much faster.

Matthew Stubenberg: Kind of getting to the point of the whole thing is in those types of situations, you might need to use something like Phantom JS which more mimics a human using a web scraper and can actually wait until everything is done downloading from that web page and then cycle through it a little bit easier.

Matthew Stubenberg: I find Phantom JS a little bit more tedious to use. I think the added delay time can be a real factor if you're trying to pull millions of cases.

Matthew Stubenberg: Okay, so now I've gone over all the great different, or actually, sorry I think I slipped a slide here. Okay, so the other big thing that we use web scrapers for here, is to look up client information.

Matthew Stubenberg: Let's say you're a client and you applied to MBLS for custody help, right? You need an attorney to represent you in a custody battle. Well we, at MBLS, we help with almost every civil legal issue. What we do is after you filled out our MBLS form and that data gets put into our system, I've built a program that will automatically use all of these web scrapers to go out and look for other pieces of information you might have that kind of raise a red flag that might indicate that you have some other legal issues that need to be taken care of.

Matthew Stubenberg: For example, we'll take your name and date of birth and search to see if you have any cases eligible for expungement. We'll see if you have a water bill that's above a certain amount to kind of trigger a warning to see if we can get you on a payment plan before it goes to tax sale. We check for potential tax credits. If you fall into certain income brackets and you own your house in certain counties, you can apply for certain housing tax credits and we can check to see if you've already applied for them. We can check to see if you have other cases that you might not know about.

Matthew Stubenberg: You come in for your custody suit that you're trying to get help with, but then while you're here, while we actually have you in front of us, we also help you with getting your cases expunged. We help challenge your high water bill. We get you some beneficial tax credits and we inform you that, by the way, you're also being sued for some tort violation or some unpaid bill. Did you know about that? Do you have an attorney? Are you taking care of it or do you need help with that as well?

Matthew Stubenberg: It's a very interesting cool tool that kind of takes a holistic approach to helping the client. We have a number of different ways that we can alert different people to it, right? If you find something that's really pressing you can send a text message to the client saying hey we found this, do you need help? Text back yes I need some help or we can raise a flag with the attorney for the
attorney to look into it a little bit deeper, or just simply make a case note of it for later analytical purposes.

Matthew Stubenberg: It's been very helpful in trying to catch some of these outlying legal issues that, unfortunately, a lot of our clients tend to have.

Matthew Stubenberg: Okay, so hopefully I've convinced everybody that web scraping is the way to go. You want to be careful not to get on the wrong side of the law, which can sometimes be easier than you might think.

Matthew Stubenberg: The first thing I want to talk about is a DOS attack. What you're probably familiar with is a DDOS attack, which is what this is a picture of right here, the extra D stands for distributed denial of service, but when it's just one particular computer, which is making so many requests to a particular server that the server gets overwhelmed and can't handle requests from other legitimate traffic, that's called a DOS attack.

Matthew Stubenberg: What might cause so much traffic? Well, somebody using a web scraper. I will say that most servers, anything that's close to professional, should be able to handle as much traffic as one single computer can throw at it. However, it's something to take into consideration if what you're trying to scrape is a mom and pop website that might not be built that well or is fairly old or some other website that maybe is built okay but gets so much traffic that the extra burden on the server might be too much. The last thing you want to do is cause some kind of harm to the server that's hosting the website and stop legitimate people from getting there.

Matthew Stubenberg: How do you take this into consideration? There's a number of different ways. One you could always call the people hosting the website, although most people don't tend to take kindly to you trying to download their whole database. Another way is you can, what kind of website is it? Who hosts it? Is this a state run website that should have the infrastructure that'll hold it or is it mom and pop? What's important is you can add a time delay in between your scraping.

Matthew Stubenberg: For instance, with the property scraper, when we scraped out all the water bill data, we added a time delay of a second in between each search in an attempt to not overload their system. We didn't want to stop other people from looking up their water bills. We basically said, alright web scraper, after you search this address, pause for one second and then search the next one. That's a good way of making sure that you don't try to overload the system. That's what Google recommends as their recommended time in between scraping. If Google comes to your site and scrapes your site it'll do it at one second per page.

Matthew Stubenberg: The next big thing is Terms of Use. The legal area here is very gray. Whether a website can prevent somebody from using a web scraper through a Terms of Use kind of is a legal gray area depending on how you're using it, whether you're overloading their system, how prominent the Terms of Conditions are
displayed. There's a number of factors that go into it, so I don't want to give you legal advice on that, but my recommendation is simply don't violate the Terms of Use.

Matthew Stubenberg: If they say you're prohibited from using robots, web scrapers, anything like that, I would usually respect that request. The last thing, even if, maybe you're legally in the right, you don't want ten years of legal court cases to figure that out. This is a screenshot of a website here in Maryland that prohibits robot collection and web scraping. It's displayed very prominently so we've never scraped them. It's a very good example of what these look like.

Matthew Stubenberg: This is CAPTCHA's and this kind of goes back to the same question that we had before. The CAPTCHA's are a pretty good sign that they don't want you to scrape their website. CAPTCHA'S are images that are very difficult for computers to read and therefore stops web scrapers from using a website. While there are ways to get around CAPTCHA'S, there's all kind of tools to be able to read what that word is and type it in or at least make a number of different guesses and you get 75% success rate or something like that, the same kind of legal issues come up. The website is making it fairly clear that they don't want to be scraped and you're doing it anyways.

Matthew Stubenberg: If you come across this, I wouldn't scrape it. The other thing is authorized access. You know, in my mind, public information is fair game that anybody can go to, but if you find a security vulnerability, you don't want to then say oh this is great, I can access all the information for all the accounts because they didn't realize that you can do X, Y and Z and figure out somebody else's account information.

Matthew Stubenberg: Unauthorized access to a computer will get you in trouble under the Computer Fraud and Abuse Act, so you want to stick to what they make very easy and obvious that people can access. It's just something to think about that even though you might have the technical ability to do it, it doesn't necessarily mean that you should do it.

Matthew Stubenberg: Hopefully everyone is now ... I didn't scare too many people away from it and everyone's on board for building their own web scraper. There's a couple of ways to do it. If you don't have any coding experience, there are a number of online user friendly tools that can help you build a web scraper without any actual code. There's things like webscraper.io.

Matthew Stubenberg: I'm always a little bit wary of these. I think these are great tools if you kind of want to just test out what a web scraper is to get a better understanding and see what goes into building it. As with anything, using a kind of nice, friendly interface is going to limit how complex and detailed a web scraper can be. I don't know if I'd use it for like an industrial actual web scraping tool. It can be a good place to go to learn how web scrapers work.
Matthew Stubenberg: The DIY method, right, is if you're motivated now to build your own web scraper, you need a working knowledge of HTML to be able to figure out how to parse through some of that information to [inaudible 00:58:03] or to store the information in a database. Then you need some kind of programming language like Python, which is what I tend to use most frequently for web scraping. Then, depending on the website, you might want to learn something like JavaScript so you can use something like Phantom JS which is a headless browser.

Matthew Stubenberg: If everything I've said went over your head and you're one of those people who say this is simply not for me, there are lots of people who already have this knowledge conveniently. You can use websites like Freelancer to find a freelancer who could build you these kind of things if you search in your search terms for things like web scraping and data mining, you can find different people that have those skills. I've screen shotted just somebody who had these skills. This is a perfect example of the skills that you need in order to be a good web scraper. This person has a background knowledge in PHP and Python. They understand HTML. They know MySQL so they could store it all. Not only that they have experience in actually web scraping and crawling. This person is only $30 per hour, which is almost too good to be true because that's a really good price for someone who knows all of this and has many years all of this experience.

Matthew Stubenberg: There are ways to find people to build these small projects. That's pretty much the end of my presentation. If you're curious to, if you have a website that you're thinking about scraping, or some questions about web scraping or how I built some of the projects, especially if you have the difference between Python or Phantom JS and things like that, that may be slightly more complex than what this presentation was for, feel free to email me. I love talking about this stuff. It's not going to be a burden at all.

Matthew Stubenberg: While I might not be able to build the project for you, I'd be happy to take a look at a website to see whether this is going to be something extremely hard or really easy to do.

Erin Farley: Great. Matthew, thank you very much. Before you go, can you, the last thing I want to do is allow people to provide some feedback, if you could drag that, I like to call it the Magic Ball, over to my name, I'm going to launch a poll. Oh, Jason grabbed it, so Jason's going to launch the poll. There it is. Then if everybody who's still here could fill this out and give us some feedback, we would greatly appreciate it. Matthew, thank you again so much for this webinar, it was really interesting. We will post this and have this up tomorrow. Thank you everyone.

Matthew Stubenberg: Fantastic, thanks for having me.